🎨

디자인 패턴

컴포지트(Composite)

트리(Tree) 구조를 만들 때 사용하는 패턴으로, 개별 객체와 그룹 객체를 동일하게 다루도록 도와준다. 복합체 패턴이라고도 한다.

!? 언제 사용할까?

- 트리 구조를 사용해야 할 때
- 개별 객체와 그룹 객체를 동일한 방식으로 다뤄야 할 때
- 객체 간의 계층 관계(부분-전체 관계)가 필요한 경우

🏗️ 컴포지트 패턴 구조

Component (컴포넌트, 인터페이스/추상 클래스)

- Leaf와 Composite이 공통적으로 가져야 할 기능을 정의

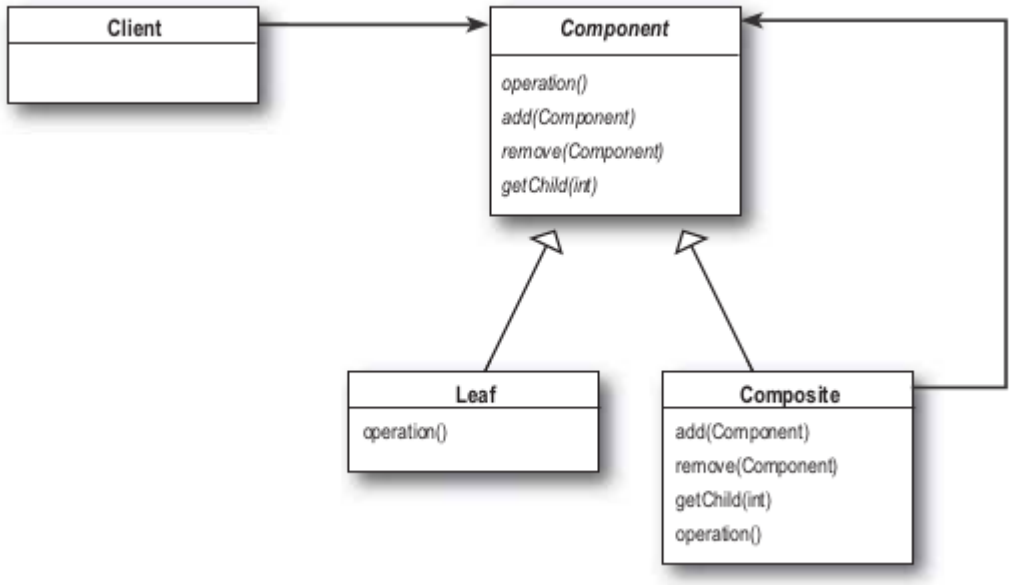
Leaf (잎/단일 객체, 개별 요소)

- 더 이상 하위 요소를 가질 수 없는 객체

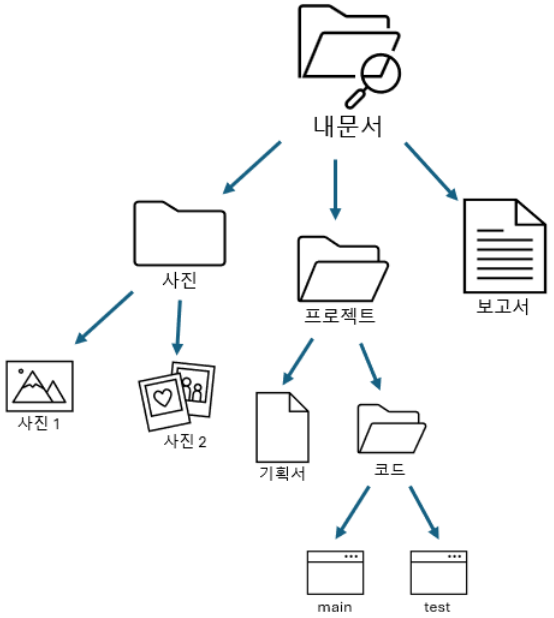
Composite (복합 객체, 그룹 요소)

- 여러 개의 Leaf 또는 다른 Composite을 포함하는 객체

🌐 UML



📌 적용 예시



이 파일 시스템은 **트리 형태**로 계층적 구조를 이루고 있다.

1. 최상위 폴더: 내문서

- 모든 파일과 폴더의 최상위 요소(루트)이다.
- 이 폴더는 다른 폴더(Composite)와 파일(Leaf)을 포함한다.

2. 1단계 하위 요소

- **사진 폴더**: 하위에 두 개의 사진 파일(사진 1, 사진 2)을 포함.
- **프로젝트 폴더**: 하위에 기획서 파일과 코드 폴더를 포함.
- **보고서**: 더 이상 하위 요소가 없는 단일 파일.

3. 2단계 하위 요소 (중첩된 Composite)

- **코드 폴더**: main과 test라는 파일(Leaf)을 포함.
- 사진 1, 사진 2, 기획서, main, test는 모두 하위 요소를 가지지 않는 단일 파일(Leaf)이다.

👍 컴포지트 패턴의 활용 장점

1. **유연한 구조 확장**: 새로운 폴더나 파일을 쉽게 추가/삭제 가능.
2. **일관된 인터페이스**: Composite와 Leaf를 동일한 방식으로 다룰 수 있음.
3. **재귀적 탐색**: 상위 Composite에서 하위 요소를 재귀적으로 처리 가능.

어댑터(Adapter)

어댑터는 호환되지 않는 인터페이스를 가진 객체들이 협업할 수 있도록 하는 구조적 디자인 패턴이다. 쉽게 말해, "인터페이스가 맞지 않는 클래스들을 연결해주는 변환기"라고 생각하면 된다.

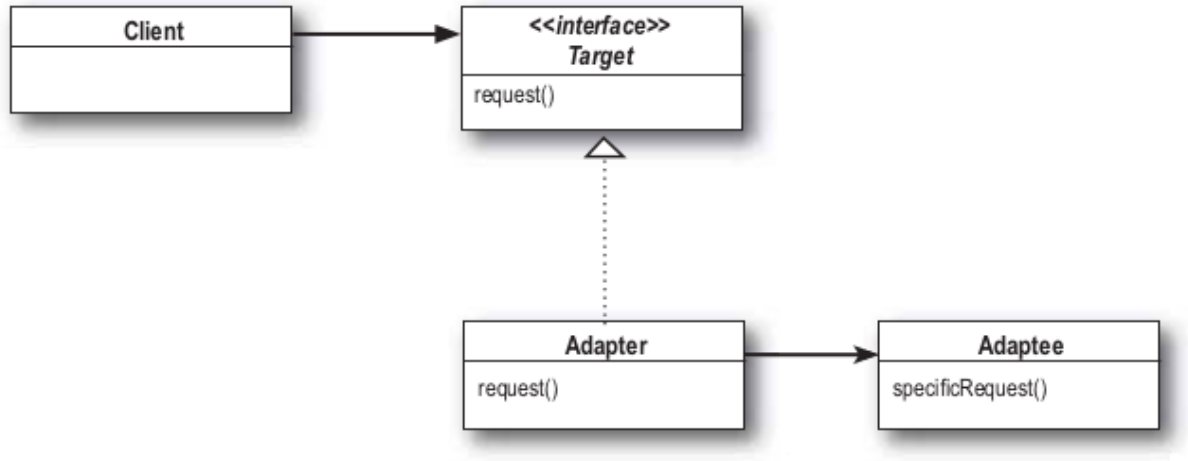
🎯 어댑터 패턴이 필요한 이유

1. **기존 코드 재사용**: 이미 존재하는 클래스나 라이브러리가 있지만, 인터페이스가 다르면 재사용이 어려움.
2. **새로운 시스템과의 통합**: 다른 시스템의 클래스를 수정하지 않고 내 코드와 통합하고 싶을 때.

🏗️ 어댑터 패턴 구조

1. **클라이언트(Client)**: 어댑터를 통해 타겟 인터페이스를 사용하는 주체.
2. **타겟(Target)**: 클라이언트가 기대하는 인터페이스.
3. **어댑터(Adapter)**: 타겟 인터페이스를 구현하면서, 실제 사용할 어댑티(Adaptee)의 기능을 변환해 연결.
4. **어댑티(Adaptee)**: 기존 클래스 또는 변환 대상.

🌐UML



📌 어댑터 패턴의 실제 활용 사례

1. **외부 라이브러리 통합**: 내가 사용하는 코드와 외부 라이브러리의 인터페이스가 다를 때.
   - 예: JSON을 사용하는 코드와 XML 데이터를 반환하는 API 통합.
2. **GUI 툴킷**: GUI 위젯을 새롭게 정의하거나, 플랫폼별 차이를 통합.
3. **데이터베이스 연동**: ORM(Object Relational Mapping)에서 데이터베이스 인터페이스를 통합.
4. **게임 개발**: 다른 물리 엔진이나 그래픽 엔진을 게임 엔진에 연결.

💡 비유

- 220v 한국 플러그(Adaptee)와 110v 미국 소켓(Target)이 호환되지 않는다.

- **어댑터**가 중간에서 둘을 연결해 준다.
- **결과**: 아무 문제 없이 한국 플러그를 미국에서 사용할 수 있게 된다.

👍어댑터 패턴의 장점

1. **호환성 제공**: 기존 클래스나 코드를 수정하지 않고도 새 환경에 맞게 사용할 수 있음
2. **재사용성 향상**: 기존 코드를 재활용하면서 새 기능을 추가 가능
3. **유연한 설계**: 시스템 간의 통합 및 확장이 용이